1/90

by Bill Sefton

APPLE DISK & MEMORY UTILITY

- Repairs Blown Disks
- Reads Nibbles
- Maps Disk Space
- Searches Disks
- Searches Memory
- Edits Disk Sectors
- Outputs Screen to Printer
- Displays Memory In HEX/ASCII
- AND MUCH MORE

Omega Software Products, Inc
222 S. Riverside Plaza, Chicago, IL 60606
(312) 648-1944

ACCESSING THE INSPECTOR

A. APPLE II STANDARD (Integer on mother board).

FROM	TO	TYPE
Integer	Inspector	CALL —10240, (RETURN)
Monitor	Inspector	C081, (RETURN) D800G, (RETURN).
Applesoft	Inspector	CALL —151, (RETURN), C081, (RETURN), D800G, (RETURN)

B. APPLE II+ (Applesoft on Mother Board)
(Integer on Firmware Card)
(Integer in Ram of Language Card)

FROM	TO	TYPE
Integer	Inspector	CALL —10240, (RETURN)
Monitor	Inspector	C080, (RETURN), D800G, (RETURN)
Applesoft	Inspector	CALL —151, (RETURN), C080, (RETURN), D800G (RETURN)

FROM	TO	TYPE
Integer	Inspector	CALL —10240, (RETURN)
Applesoft	Inspector	& (RETURN)
Monitor	Inspector	CTRL—Y, (RETURN)

REFERENCE CHART

COMMAND	CODE	PAGE
1. ASCII/Hexdecimal display mode	A	10
2. Buffer Location	B	11
3. DOS Version	V	9
4. Drive Number	D	9
5. Edit	E	10
6. Exchange	X	10
7. Return to Basic	C	18
8. Find	F	16
9. Hex/ASCII Dump	H	16
10. Incremental Sector and Buffer	CTRL—I	12
11. Locate	L	17
12. Map	M	13
13. Nibble read	N	13
14. Peripheral Slot	P	9
15. Read	R	9
16. Read Nibbles For Next Track	^	15
17. Scan Backward	—	10
18. Scan Forward	.	10
19. Sector	S	9
20. Track	T	9
21. Write	CTRL—W	11
22. User Command	CTRL—Z	18

Omega Software Products makes no warranties, either expressed or implied, with respect to this product, its quality, performance, merchantability, or fitness for any particular purpose or use. This product is sold 'as is'. Omega Software Products, Inc. will not be liable for direct, indirect, incidental or consequential damages as a result of the use of this product or as a result of any defect even if we have been notified of the possibility of such damages. This product, its media and documentation are copyrighted. Duplication is permitted only for archival purposes. Distribution for any other purpose, whether for profit or not, is a violation of federal law.

©1981 Omega Software Products, Inc.
Apple is a registered trademark of Apple Computer, Inc.

TABLE OF CONTENTS

PAGE

A. General Description 1

B. System Requirements 1

C. Installation 2

1. Installing THE INSPECTOR (Firmware version) 2
2. Installing THE INSPECTOR (Software version) 4

D. Accessing THE INSPECTOR 5

E. The Display 6

F. Using THE INSPECTOR 8

1. Disk Routines 9
2. Memory Routines 16
3. Miscellaneous Commands 18

G. Tips For the User 19

1. Foreign DOS 19
2. Finding RWTS 19
3. Using A Printer 20
4. Reading Half Tracks 21

H. Applications 22

1. Verifying Disks 22
2. Transfer DOS 22
3. Eliminate Control Characters 22
4. Undelete Programs 23
5. Editing BASIC Programs 24
6. Avoiding the Reloading of the Language Card 25

I. Appendix

1. A Bit About Nibbles 26
2. Suggested Reading 28
3. Decimal to Hexadecimal Conversion Chart ..... Inside Back Cover

The Inspector

Personal Service Department

(312) 648-1715

You may call our Personal Service Department during normal business hours (note: we are located in the central time zone) for help in using this product. Please, no collect calls! Our staff will try to solve your problems while you are on the phone. Questions regarding usage sent by letter will be answered also. Please put your return address on the letter itself.

Thank You

GENERAL DESCRIPTION

THE INSPECTOR is a powerful, simple to use utility, which allows you to examine and edit data that is either in the APPLE's memory or on APPLE II DISKS. You can even examine and change information on disks which have non-standard DOS.

THE INSPECTOR is a permanent-memory device that is plugged into the APPLE. This means that it can be used at any time without stopping to load an extra program. It uses part of the APPLE'S unused memory allocation so it won't interfere with other programs in memory.

THE INSPECTOR is primarily designed to allow you to read information from a disk, edit and change this information and write it back to the disk. It can quickly scan through a disk, one sector at a time. Its simple commands allow you to specify the track and sector which you wish to examine. The information may be displayed, either as alphanumerical or hexadecimal data, at your command.

These features make THE INSPECTOR an invaluable aid when working with the APPLE II DISK. Blown disks can often be recovered, catalog entries fixed, control characters deleted and specific data found on disks. It is also very helpful in finding and repairing bad files in data bases, even when the data disk is not in normal DOS format.

THE INSPECTOR is not only limited to working with disks. It also can be used to search through and examine all the information in the APPLE's memory. It will allow you to look for a particular HEX or ASCII string or display a HEX/ASCII memory dump starting at any location in memory. You may also edit the information that you find in memory without exiting the THE INSPECTOR.

SYSTEM REQUIREMENTS

THE INSPECTOR is compatible with any APPLE II STANDARD computer. You will need an INTEGER CARD, APPLE LANGUAGE CARD or a 16K RAM CARD if you have an APPLE II+. You will also require at least one APPLE DISK II DRIVE for the disk routines. THE INSPECTOR supports lower case adapters and will operate with as little as 16K of RAM memory.

INSTALLATION

WARNING: BEFORE ATTEMPTING TO INSTALL THE INSPECTOR EPROM, TURN OFF THE POWER TO THE APPLE TO PREVENT DAMAGING YOUR SYSTEM.

INSTALLATION OF THE INSPECTOR (EPROM VERSION)

(If you have THE INSPECTOR disk version, skip to the following section titled Installation of the Inspector (disk version))

THE INSPECTOR EPROM is designed to be inserted in the D8 socket on the motherboard of an APPLE II or in the D8 socket on the INTEGER CARD on an APPLE II+.

INSTALLING THE INSPECTOR IN THE APPLE.

1. Double check that power is off.
2. Remove the cover from the computer.
3. Touch the case of the power supply to discharge static electricity that may have accumulated on your body.

IF YOU HAVE A STANDARD APPLE II

4. Locate socket D8 on the motherboard.
5. Orient the EPROM so the notch is facing towards the key board. See figure 1.

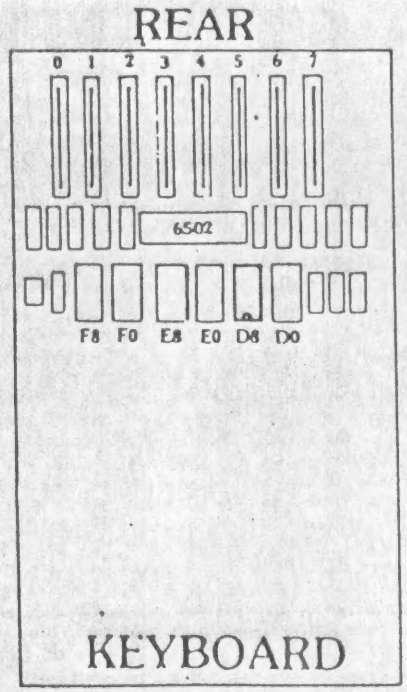

Figure 1

6. Now carefully align the pins with the socket and gently but firmly insert the EPROM into the D8 socket until it is seated. You may find that the pins on the EPROM need slight bending to get them properly aligned. They can be bent quite easily by carefully pressing them against a flat surface.

IF YOU HAVE AN APPLE II+ (with Integer firmware card)

4. Remove the INTEGER FIRMWARE CARD from slot #0.
5. Locate socket D8 on the INTEGER FIRMWARE CARD.
6. Orient the EPROM with its notch towards the top of the INTEGER FIRMWARE CARD. See Figure 2.

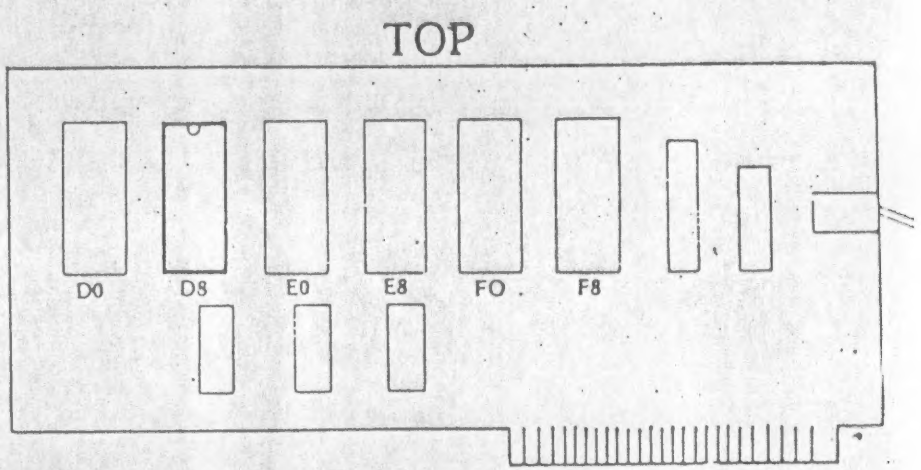

Figure 2

7. Now carefully align the pins with the socket and gently but firmly insert the EPROM into the D8 socket until it is seated. You may find that the pins on the EPROM need slight bending to get them properly aligned. They can be bent quite easily by carefully pressing them against a flat surface.
8. Insert the INTEGER FIRMWARE CARD back into slot #0.

The power to the APPLE can now be turned on and THE INSPECTOR is always available for use.

INSTALLATION OF THE INSPECTOR (Disk Version)

THE INSPECTOR is also supplied on disk for users who have an Apple II+ with either the Apple Language system or a 16K RAM memory expansion board. The disk version is installed as follows:

1. Boot DOS into the Apple using a BACKUP COPY of your DOS 3.3 System Master Diskette.
2. Insert THE INSPECTOR disk into the disk drive. Type 'RUN THE INSPECTOR', (RETURN).
3. You will see the following:

   DO YOU WISH TO MODIFY

   1. DOS 3.3 BASICS
      OR
   2. DOS 3.3 SYSTEM MASTER

   NOTE: You will be modifying these disks so we suggest that you use BACKUP COPIES.

   Press (RETURN) after inserting the correct disk.

   Your disk will be modified and the screen will say:

   YOUR DISK HAS BEEN MODIFIED AND CONTAINS THE INSPECTOR.

4. Place an Inspector sticker on the disk for identification.
5. THE INSPECTOR may be accessed and used in the same fashion as the EPROM version.

THE INSPECTOR is loaded into the Apple every time you boot the BACKUP COPY of your DOS System Master disk or BASICS disk.

ACCESSING THE INSPECTOR

THE INSPECTOR is always in memory and available for use. THE INSPECTOR can be used in a number of different APPLE configurations. Since each is slightly different there are a number of distinct procedures to be used depending on the computer configuration you are using. To access THE INSPECTOR proceed as specified for the Apple configuration which you are using.

A. APPLE II STANDARD (Integer on mother board).

FROM	TO	TYPE
Integer	Inspector	CALL —10240, (RETURN)
Monitor	Inspector	C081, (RETURN) D800G, (RETURN),
Applesoft	Inspector	CALL —151, (RETURN), C081, (RETURN), D800G, (RETURN)

B. APPLE II+ (Applesoft on mother board) (Integer on Firmware Card or in RAM of Language Card)

FROM	TO	TYPE
Integer	Inspector	CALL —10240, (RETURN)
Monitor	Inspector	C080, (RETURN), D800G, (RETURN)
Applesoft	Inspector	CALL —151, (RETURN), C080, (RETURN), D800G (RETURN)

WARNING: GOING DIRECTLY FROM APPLESOFT TO INTEGER WILL DESTROY THE APPLESOFT PROGRAM WHICH IS ALREADY IN MEMORY. TO PREVENT THIS PROBLEM, WE GO FROM APPLESOFT TO MONITOR AND THEN ACCESS THE INSPECTOR.

Once you have initialized or used the INSPECTOR you can easily go from a program to THE INSPECTOR.

FROM	TO	TYPE
Integer	Inspector	CALL —10240, (RETURN)
Applesoft	Inspector	& (RETURN)
Monitor	Inspector	CTRL—Y, (RETURN)

THE DISPLAY

When the utility is accessed you will get the following display.

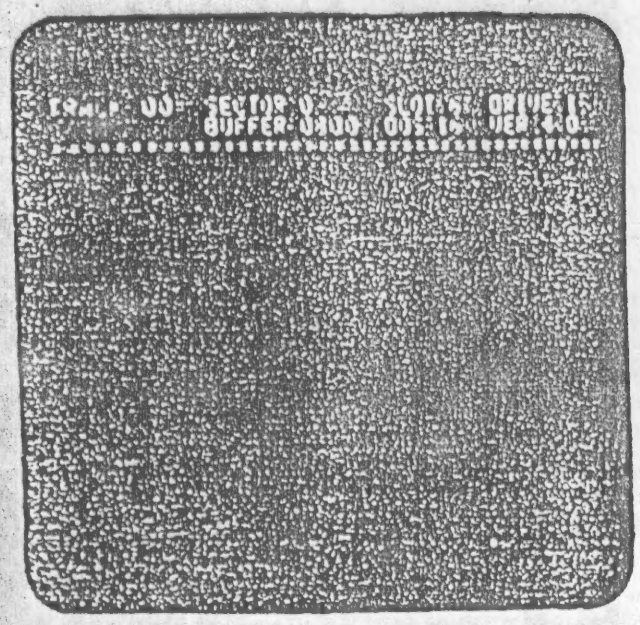

Figure 3.

After the utility is accessed, the screen will blank and the top 2 lines will display the utility status. Figure 4 is an example of the display after it is accessed and the 'A' key is pressed.

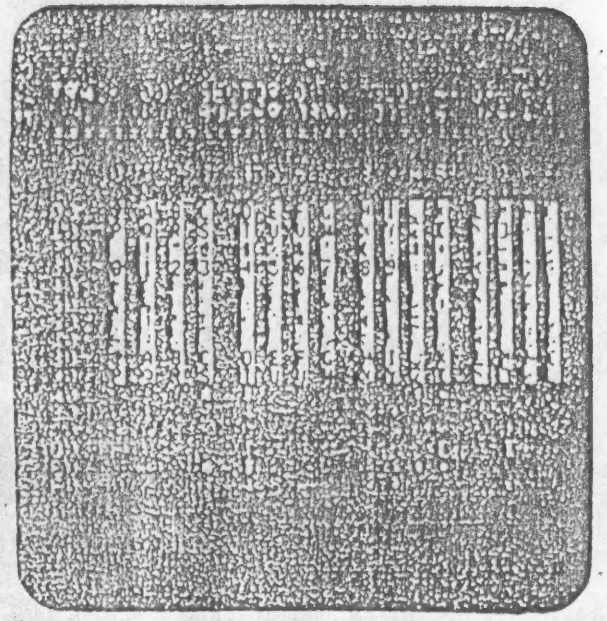

Figure 4. The buffer was filled with values $00 through $FF to show the displayed ASCII values for the utility.

Notes:

1. The VERSION number of THE INSPECTOR is only displayed at the start of the utility, and will blank after the first READ or WRITE command.

2. A specific byte in the buffer may be located by adding the values of the column and row that the byte appears in. For example to find byte $DF, you would look in row $D0 and column $F. By adding this value to the BUFFER value, (in this case $0800) the byte location in memory is found. Hence, this is byte $08DF.

3. This utility displays all ASCII values as they should appear. The only exceptions to this are control characters which are displayed in inverse. Bytes $00 through $3F contain the full ASCII character set in inverse, and bytes $40 through $7F contain the ASCII set in flashing mode. Bytes $80 through $9F are the control characters but to avoid problems are displayed as inverse characters. Bytes $A0 through $DF are the normal ASCII characters and bytes $E0 through $FF are lower case. THE INSPECTOR supports lower case as shown in figure 4, but only when there is a lower case adapter installed in the APPLE.

USING THE INSPECTOR

THE INSPECTOR is designed to be both simple and versatile to use. It recognizes 22 different commands. To get the desired function, press the key shown before each command.

A reference will be made to commands that can be toggled. This simply means that the command has two settings and you may go from one to the other by pressing the designated key. For instance when THE INSPECTOR is selected it defaults to Disk Drive 1. You may change to Drive 2 by pressing 'D'. If you press 'D' again your selection will toggle back to Drive 1.

DISK ROUTINES

V = DOS VERSION

Selects the DOS VERSION to match the version on the disk that you are using (default is 16 sectors). The selection choice can be toggled between 13 or 16 sectors to be compatible with the disk you are working on.

NOTE: This is the first thing that you should check when using THE INSPECTOR.

P = PERIPHERAL SLOT

Selects the PERIPHERAL SLOT number where the disk controller card is located. (Default is slot #6. If another slot is being used, enter its correct number by entering; P,slot number, (RETURN).

D = DRIVE NUMBER

Selects the DRIVE NUMBER you want to work with (default is Drive 1). Toggles between drives 1 and 2.

T = TRACK

Selects the TRACK number on the disk which you wish to read.
To use this function:

1. Press 'T'. The cursor will move to the Track portion of the display.
2. Enter the Track number you wish to read and press (RETURN).

S = SECTOR

Selects the SECTOR you wish to read (you may select 0—C for a 13 sector disk and 0—F for a 16 sector disk). To use this function:

1. Press 'S'. The cursor will move to the Sector portion of the display.
2. Enter the Sector number that you want and press (RETURN).

NOTE: ALL TRACK, SECTOR, AND BUFFER INFORMATION MUST BE ENTERED IN HEXADECIMAL NOTATION. FOR YOUR CONVENIENCE A DECIMAL TO HEXADECIMAL CHART HAS BEEN INCLUDED ON INSIDE REAR COVER.

NOTE: WHEN WORKING WITH A 16 SECTOR DISK, DOS VERSION 16 MUST BE SELECTED OR YOU WILL NOT BE ABLE TO ACCESS SECTORS D—F.

R = READ

Causes the utility to READ a sector of information from the disk according to the track and sector which you have specified.

Errors encountered during the read process will be displayed in the upper right hand portion of the screen. This display will clear upon reading a good sector.

; = SCAN FORWARD (Unshifted + Key)

Causes THE INSPECTOR to SCAN FORWARD one sector. When it reaches the last sector on a track (C on a 13 and F on a 16 sector disk respectively) it will increment to the next higher number track. This command makes it very convenient for you to scan through a disk to see what is on it. This is especially true if you also use the REPEAT key.

Each sector will be read into the same buffer location. Hence, when a new sector is read in, it will replace the old sector in the buffer location designated.

— = SCAN BACKWARD (minus key)

Causes THE INSPECTOR to SCAN BACKWARD one sector. It is the reverse of SCAN FORWARD but performs similar functions.

A = ASCII/HEXADECIMAL DISPLAY MODE

Toggles between the ASCII and HEXADECIMAL display mode. Causes the contents of the buffer to be displayed in either ASCII or HEXADECIMAL notation.

X = E(X)CHANGE

EXCHANGES flashing and inverse ASCII characters with normal ASCII characters. When there are many flashing or inverse characters on the screen at one time they can be difficult to read. This command acts as a filter to convert them to a normal display that is more easily read.

E = EDIT

Selects the EDIT mode so that you can edit (change, insert or delete) information in memory from either the Hexadecimal or ASCII display mode.

This command has a few variations depending on the mode you are using so note the differences carefully and you won't go wrong.

For Hexadecimal notation use it as follows:

1. Toggle the ASCII/Hexadecimal display mode to the Hexadecimal mode by pressing "A".
2. Type "E" to select the Edit Mode.
3. Type the number of the byte location you wish to change.
4. Type (SPACE). The current contents in that byte location will be displayed in the upper left hand portion of the screen.
5. Type in the new hexadecimal information that you wish to place in that byte location.
6. Press (RETURN) to transfer this new information to the byte location that you have specified.
7. If you wish to change successive byte locations press, (SPACE BAR) instead of (RETURN). Press (RETURN) to exit.

Editing ASCII, Negative ASCII (flashing ASCII) or lowercase letters is slightly different. The procedure is as follows:

1. Toggle the ASCII/Hexadecimal mode to the ASCII mode by pressing "A".
2. Type "E" to select the Edit mode.
3. Type the number of the byte location you wish to change.
4. Type "L" if you want to insert lowercase letters or type "N" if you want to insert negative ASCII characters or the (SPACE BAR) for normal ASCII.

NOTE: Some ASCII data must be stored with the most significant bit turned off (it will appear as flashing ASCII characters) we have called this negative ASCII.

5. Enter the ASCII string into the desired locations.
6. Press (RETURN) to exit from the EDIT MODE.

CTRL—W = WRITE

Accesses the drive which you have selected and writes the contents of the selected buffer onto the disk at the track and sector location you have selected.

WARNING: USE CAUTION WHEN WRITING TO THE DISK. IF THE DATA IN THE BUFFER HAS BEEN EDITED OR THE TRACK, SECTOR, OR BUFFER NUMBER HAS BEEN CHANGED, IT IS POSSIBLE TO WRITE INCORRECT INFORMATION TO THE DISK OR WRITE THE INFORMATION IN THE WRONG LOCATION. ALWAYS DOUBLECHECK BEFORE WRITING TO A DISK!!!

B = BUFFER LOCATION

Allows you to select the starting BUFFER LOCATION where information will be stored in memory when it is read from the disk. For instance, if the buffer is set to $0800 (the default value), whenever a sector is read into memory it will be deposited in memory beginning at location $0800 and going to $08FF. This can be very convenient if you want to read a disk without overwriting a program that is already in memory. All you have to do is select the starting Buffer Location to begin at an unused portion of memory.

You may also use this command to examine memory in the APPLE as follows:

1. Press "B" to select the buffer location mode.
2. Select the buffer page you wish to see.
3. Press (RETURN). The display will then show the contents of this location in memory. Any READ or WRITE commands will now use this buffer address.

NOTE: YOU CAN ONLY SELECT THE BEGINNING OF A PAGE IN MEMORY (FOR EXAMPLE $0800, $0900... $BF00). THEREFORE ONLY THE FIRST 2 DIGITS CAN BE CHANGED WHEN USING THIS COMMAND.

NOTE: READING INFORMATION INTO BUFFER LOCATION $0000 WOULD HAVE UNPREDICTABLE RESULTS. TO PREVENT THIS, THE READ AND MAP COMMANDS ARE DISABLED IF THE BUFFER IS SET TO $0000.

CTRL—I = INCREMENT SECTOR & BUFFER

INCREMENTS both the SECTOR & BUFFER sequentially.

This mode is very convenient when you wish to read and write sequential sectors of a disk. You may do this as follows:

1. Set the number of the track and sector, (using the "T" and "S" commands) to the location on the disk where you wish to start reading.
2. Enter the buffer starting location (usually $0800) with the "B" command.
3. Enter "R" to select the READ mode. This will read the first sector.
4. Press "CTRL—I" to increment the sector and buffer numbers and to read the new sector information into the new buffer.

By using the repeat key with CTRL—I, it is possible to quickly read sequential sectors into memory. When you are finished the reading sequence, it is important to jot down the number of the last track and sector read.

To transfer these tracks to another disk proceed as follows:

1. Reset the track number to its original value using the "T" command.
2. Reset the sector number to its original value using the "S" command.
3. Reset the buffer location to its original value be using the "B" command.

MAKE SURE YOU HAVE THE DISK, THAT YOU ARE TRANSFERRING DATA TO IN THE PROPER DRIVE.

4. Type CTRL—W to select the WRITE mode. This will write the first sector.
5. Typing CTRL—I will continue writing successive sectors to the disk. When using the REPEAT key be very careful that you do not write beyond the last track and sector numbers (that you carefully jotted down) when you were reading the data into memory.

The above procedure allows you to quickly transfer entire portions of data from one disk to another (To transfer DOS for instance).

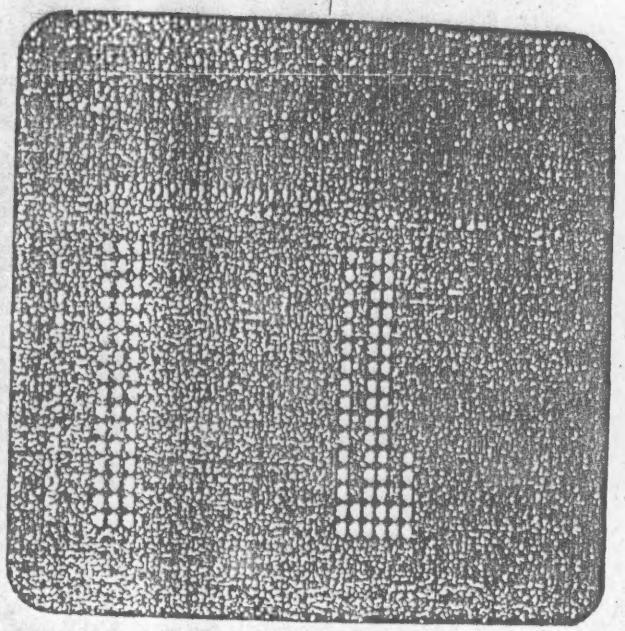

Figure 5

M = MAP

Track $11, sector $00 will be read. This contains the VTOC (Volume Table Of Contents). The display will clear below line 3 and a map will appear on the screen. "#" indicates that a sector is used and " " indicates that a sector is unused. Any command given to the utility will return to the normal display.

N = NIBBLE READ

Selects the NIBBLE READ mode which is composed of two functions. The first is a Hi Res Screen graphics display, the second is a nibble display mode. This command is very useful for troubleshooting disks. Sometimes it is not possible to read sectors into memory even when you suspect that there is valid data contained on a track. Use this function as follows:

1. Select the track number you wish to read by using the "T" command.

2. Press "N" to select the NIBBLE READ command. The track will be read and shortly thereafter, the display will switch to the Hi Res SCREEN and a graphics display of the nibbles on the track will be shown.

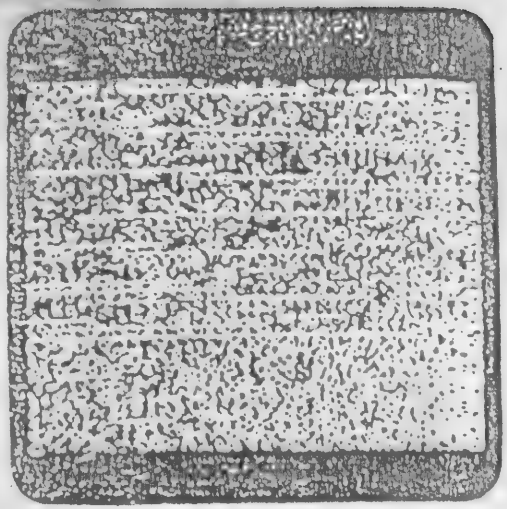

Figure 6

Figure 7

At first this display may appear confusing, but with a little practice, it is possible to distinguish between 13 and 16 sector disks or to tell how much data is on a disk. The FF's or SELF SYNC nibbles that appear between sectors on a track are seen as white horizontal lines on the screen. Since a 13 sector disk has more SELF—SYNC nibbles between sectors it is possible to tell whether you are working with a 13 or 16 sector disk. This does take practice but with time you will be able to extract useful information by looking at this display. Figure 6 is a 16 sector display. Figure 7 is a 13 sector display.

NIBBLE DISPLAY

After you are finished looking at the Hi—Res Screen, enter the Nibble Display mode by pressing any key.

In the nibble display mode, the screen is cleared below line 3 and a nibble dump will start scrolling.

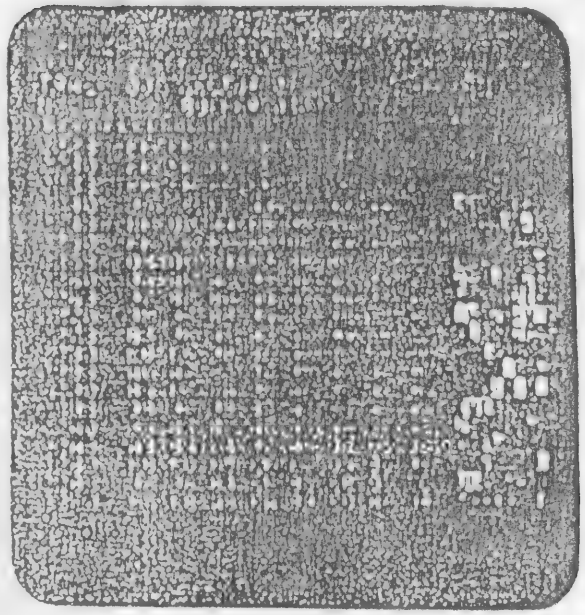

Figure 8

The display may be started and stopped by pressing any key. Pressing any key except (RETURN) will restart the display. It is very important to note that while the nibble dump shows the ASCII values for the nibbles, this is not the same as the translated byte values (see the Nibbles discussion on page 26 for further explanation). If you should accidentally scroll past some data that you wish to see, it is very simple to back up. Press any key to stop scrolling and the left arrow (←). Press any key and scrolling will continue displaying from one page back in memory. For example, if you were looking at the display at $3560, pressing the left arrow (←), will back up to $3460. Press the left arrow multiple time to back up multiple pages. While the nibble display is scrolling, pressing "Q" will toggle between fast and slow display speed. To return to the normal display, press (RETURN).

NOTE: THIS NIBBLE DUMP CAN BE A VERY POWERFUL TOOL PLEASE REFER TO THE NIBBLE SECTION PAGE 26 FOR MORE INFORMATION.

A = READ NIBBLE FOR NEXT TRACK (shifted "N")

Works like N command only the track will be incremented before doing a NIBBLE READ.

MEMORY ROUTINES

H = HEX/ASCII DUMP

Displays the HEX/ASCII information that is in MEMORY.

It is often very useful to be able to see what is in the APPLE's memory, either in HEX and ASCII mode. This can be done as follows:

1. Press "B" to select the Buffer Location mode.
2. Select the buffer location that you wish to begin at, for example $6300.
3. Press "H" to select the Hex/ASCII dump mode.

The hexadecimal and ASCII values, stored in memory, beginning at memory location $6300 will begin to scroll on the screen.

4. Press any key to stop scrolling.
5. Press any key to restart scrolling.
6. Press "Q", while scrolling, to toggle between slow and fast scrolling.
7. Press the ← (left arrow), when the display is stopped, to back up one page at a time in memory.

Press (RETURN) to exit from this mode.

F = FIND

Selects the FIND routine.

This command allows you to search all of memory (including ROMs) in the Apple for either a HEX or ASCII string. To use:

1. Press "B" to select the Buffer Location mode.
2. Set buffer page location to $0000.
3. Press "A" to select either the Hex or ASCII display mode depending on the string you wish to find.

Suppose you wish to find the string HELLO.

4. Press "E" to enter the Edit Mode.
5. Enter the length of the string in byte $00. (In our example, enter 5).
6. Enter the string for which you are searching (HELLO in our example) starting in byte $01.
7. Type "F" to start the FIND routine.

The screen will clear below line 3 and a list will appear of all the memory locations where HELLO is located.

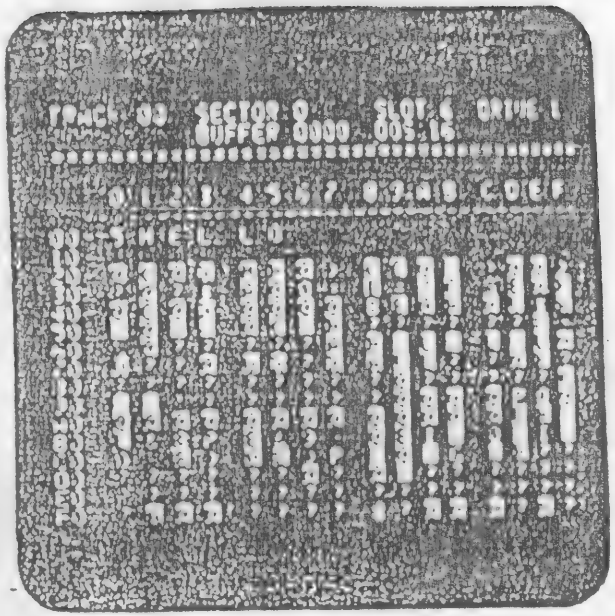

Figure 9

Pressing any key will Stop, Start the list.

Pressing any key, after the cursor appears, will return you to the normal mode.

Note: The FIND function will search for both normal and flashing ASCII character strings whenever the display is in the ASCII mode. A search in the Hexadecimal mode will find only an exact match of that string. This is also true for the LOCATE command below.

The maximum length of a string that you may search for is 9 bytes in ASCII mode, and $0F bytes in HEX mode.

WARNING: DO NOT CHANGE ANY LOCATIONS IN BUFFER PAGE $0000 ABOVE BYTE $0F. DOING THIS COULD HAVE UNPREDICTABLE RESULTS, SINCE THE APPLE USES THE LOCATIONS BETWEEN $0010 AND $00FF FOR ITS OWN POINTERS.

L = LOCATE

LOCATEs hexadecimal and ASCII strings on the disk. It is used as follows:

1. Press "B" to select the Buffer Location mode.
2. Set buffer page to $0000.
3. Press "A" to select ASCII, Hexadecimal mode display, depending on the type of string you wish to locate.
4. Press "E" to select the Edit mode.
5. Enter the length of the string in byte $00 that you wish to locate using hexadecimal or ASCII notation depending on the mode you are in.
6. Enter the string you wish to locate starting at byte $01 (using ASCII or hexadecimal notation).

Press "L". A flashing "L" will appear to the right of the Drive number on the top of the display. When a sector is read containing that string, the Apple will beep. This is very convenient when using the scan commands, since it enables the user to quickly find a particular string on a disk. THE INSPECTOR will search for the specified string. If the string is found, the scanning will pause and the Apple will beep.

As with the "F" command, the "L" will search for both normal and flashing ASCII strings when in ASCII display mode. In HEX mode, it will look for an exact match. Pressing "L" will toggle the Locate mode on and off.

MISC. COMMANDS

CTRL—Z = USER COMMAND

This command is included so that you can utilize your own subroutines. When "CTRL—Z" is pressed, it causes a jump to a subroutine at location $0300. Any subroutine can be located here, a screendump for instance. For further information, see the TIPS FOR THE USER section of this manual.

CTRL—C = RETURN TO BASIC

Pressing "CTRL—C" will return the system to the BASIC that you were using before entering THE INSPECTOR.

TIPS FOR THE USER

THE INSPECTOR was designed and written to be simple to use. Still there are many capabilities which may not be obvious to you. We will explore some of these in this section.

FOREIGN DOS

There are times when it is not possible to read from or write to a disk, because the program uses a non—standard DOS. You may find it important to be able to work with this type of disk. In the case of data bases, it is sometimes necessary to be able to work on the disk when files are lost or blown. THE INSPECTOR normally allows you to do this.

THE INSPECTOR does not use all of the DOS that is booted. It uses only the portion called Read/Write Track and Sector (RWTS), which is the section that actually does the reading and writing to a disk. Even disks that do not contain a DOS, usually have an RWTS. All you must do is find its location and then use it.

FINDING RWTS

Normal RWTS is located at $BD00 in a 48K (and $7D00 on a 32K) machine. To find RWTS on a disk with foreign DOS proceed as follows:

1. Set the buffer to $0000 using the "B" command.
2. Press "E" to select the Edit command.
3. Change bytes $00—04 to 04 84 48 85 49. This is the start sequence of almost all forms of RWTS.
4. Press the "F" key to FIND and display the occurences of this string. The highest location is the one that you want to use. Now that you have located RWTS, THE INSPECTOR can be set so that it will automatically go to this location. THE INSPECTOR normally uses the pointers at location $03D9 to point to RWTS. To get THE INSPECTOR to go to the RWTS of the foreign DOS, you must alter these pointers. This can be done as follows:
5. Set the buffer to $0300 using the "B" command.
6. Press "E" to select the Edit mode.
7. Enter D9 (which is the byte location you wish to change).
8. Type (SPACE)
9. Type 4C followed by location that you found RWTS (place the lower order byte first).

For example suppose you found the foreign RWTS at location $BD00, simply change the three locations starting at $03D9 to 4C 00 BD.

You should now be able to read and write to this disk like any other normal disk.

NOTE: IT IS STRONGLY RECOMMENDED THAT THIS BE DONE ON A BACKUP DISK, AND NOT AN ORIGINAL. THIS IS TO PREVENT ACCIDENTLY BLOWING THE ORIGINAL DISK.

USING A PRINTER

There are times when it is desirable to get a hard copy of THE INSPECTOR displays. This can be done quiet easily using the CTRL—Z users command. The CTRL—Z does a jump to a subroutine at $0300. This is convenient for letting the user define his own subroutines.

SCREEN PRINT ROUTINE

Due to the way this utility works, simply turning on the printer will not print out the Track/Sector or Map information. You must use a Screen Print routine instead.

The following Screenprint Routine may be entered from the monitor starting at location $0300 and saved to the disk. Enter the screen print routine as follows:

```
0300: A9 01 20 95 FE 20 8E FD A9 FF 20 A8 FC A2 00 8A
0310: 20 C1 FB A5 28 85 10 A5 29 85 11 A0 00 B1 10 C9
0320: A0 B0 04 69 40 D0 F8 20 3A 03 C8 C0 28 D0 EE 20
0330: 8E FD E8 E0 18 D0 D8 4C 93 FE 48 A9 00 85 24 A9
0340: 01 20 5B FB 68 4C ED FD
```

Note: Byte $0301 contains the slot # of the printer (in this case, 1). If the printer interface card echoes characters to the screen, the printout will be double spaced instead of single spaced. To prevent this, change the 3 locations $032F — $0331 to EA's. This will correct the problem. This routine may be saved to the disk and may be used in conjunction with CTRL—Z to give hard copy.

SAVING THE SCREENPRINT ROUTINE TO DISK

The author found it to very convenient to have a screenprint available when working on his disks and always keeps one handy. Rather than save a binary file on disk, he found that the routine may be saved in a sector reserved for DOS but which is not normally used. You can do the same by doing the following:

1. After entering the Screenprint Routine into memory, press "B" to select the Buffer starting location.
2. Set the buffer to $0300.
3. Set the track number to 2 and sector to F for a 16 sector disk (or sector C for a 13 sector disk).
4. Press CTRL - W to save the Screen Print routine to disk.
5. Using the "B" command, reset the buffer to its original value.

You have now saved the screen print to your disk for use at a later time.

LOADING THE SCREEN PRINT ROUTINE FROM DISK TO MEMORY

To load the Screen Print routine into memory from the disk do the following:

1. Press "B" to select the BUFFER STARTING location.
2. Set the buffer to $0300.
3. Set the track number to 2 and the sector number to F for a 16 sector disk (or sector C for a 13 sector disk).
4. Press "R" to read the subroutine from the disk into memory.
5. Using the "B" command reset the buffer to its original value.
6. Now any time you wish to print the information that is present on the screen, simply press CTRL Z 0

Due to the way THE INSPECTOR works it is not possible to use a Screen Print with the Hex/ASCII or Nibble Dump. To get hard copy of these, the following may be used.

1. Press "B" to select the starting BUFFER location.
2. Set the buffer to $0300.
3. Press "E" to select the EDIT mode.
4. Change byte locations 00—05 respectively to A9 01 20 95 FE 60.

NOTE: Byte $01 should be set to match the slot number where the printer interface card is located. In this case it has been set to 1.

Now if you are stopped in display mode during a HEX/ASCII or NIBBLE DUMP, pressing CTRL—Z will turn on the printer to print any further output. This is a very simple routine but, it is very effective. The memory dump will continue until you press RESET.

READING HALF TRACKS

Occasionally some information is placed between the normal tracks on a disk. This can make it very difficult to determine where legitimate data is located. On some disk drives, it is possible to read the data located on half tracks by reading the adjacent tracks. This is primarily a function of the read/write head characteristics. On other drives, it is very hard to determine whether or not there is legitimate data located on the half track. THE INSPECTOR makes it possible for you to read the nibbles on the half track and determine where legitimate data resides.

Under Normal DOS, the disk read head seeks the correct track by reading the track number that is specified. Then the stepper motor, which drives the head, is pulsed twice for each track. For example, if the head is on track 0 and were to be stepped until it reached track 4, the stepper motor would be pulsed 8 times. THE INSPECTOR makes it possible for you to edit 1 byte and have the utility step one pulse for every track that is specified, instead of two. Using this feature, half tracks may be reached, and inspected.

This will only work from within the INSPECTOR, since the byte that must be changed is one that is set-up in RAM by THE INSPECTOR. To access half tracks, proceed as follows:

1. Select buffer location $0200, Using the "B" command.
2. Press "E" to enter the EDIT mode.
3. In the Hexadecimal format change byte $D4 from $01 to $00.
4. Return the buffer to its previous value (usually $0800) using the "B" command.

THE INSPECTOR is now ready to read half tracks.

Since the head is set to receive one pulse for each track specified, it is now necessary to double the track number that you specify in order to move the head to each track. For example, to reach track $0 the track must be set to $0 using the "T" command, but to reach track $4, the track must be set to $8. To reach track $7, the track must be set to $E, etc. Similarly to reach a half track simply multiply by 2. For example to reach track $A.5 the track must be set to $15, for $10.5 set the number to $21.

NOTE: This feature can only be used with the NIBBLE function. RWTS normally does a check to see that the track it is reading, matches the track that was specified. In the case here, they will not match since the track specified is a doubled number. Hence, the READ or WRITE functions will not work. By using the NIBBLE function, you may these half tracks and determine if there is legitimate information present.

APPLICATIONS

Now let us explore some possible applications for THE INSPECTOR. We will look at just a few of its possible uses. These examples are meant simply to assist the user in some of the more common applications.

When working on disks with this utility, it is recommended that the user have an understanding of the diskette format. The APPLE DOS MANUAL goes into this in depth on pages 127—135. You will find other references in the Suggested Reading Section page of this manual. It is recommended that these be read to familiarize the user with this information.

VERIFY DISKS

Sometimes it is very useful to scan a disk to see what is contained on it. Using the FORWARD or REVERSE SCAN functions with the REPEAT key, a disk can quickly be examined to find what it contains. This is also very useful to help find bad sectors.

TRANSFER DOS

Occasionally DOS will be blown on a disk and all other information may be intact. In this situation all you have to do is to put DOS back on the disk without disturbing the rest of the information. This may be done as follows:

1. Insert a good disk in the drive.
2. Set the buffer to $0800, using the "B" command.
3. Press "T" and set the track to $0
4. Press "S" and set the sector to $0.
5. Press "R" to set the READ mode. This will read the first sector.
6. Use CTRL—I and the Repeat key to read in tracks $0—$2 inclusive (there may be some I/O errors at the end of track $2).

Now put this information on your blown disk as follows:

1. Insert your blown disk into the disk drive.
2. Reset the buffer to $0800 using the "B" command.
3. Press "T" and reset the track to $0.
4. Press "S" and reset the sector to $0.
5. Press "CTRL W" to set the WRITE mode. This will write the first sector. Once again use the CTRL—I and REPEAT keys to write to the disk. Do not write past track $2 or you may destroy existing data on the disk.

This method may also be used to reconstruct blown disks from their backups. After finding the bad sectors on the blown disk, they may be replaced from the backup.

ELIMINATE CONTROL CHARACTERS

Sometimes control characters are embedded in filenames either accidentally, or on purpose. Using the INSPECTOR, it is easy to eliminate these control characters.

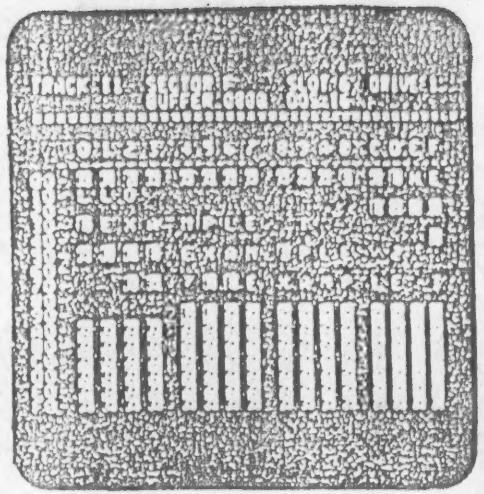

Figure 10

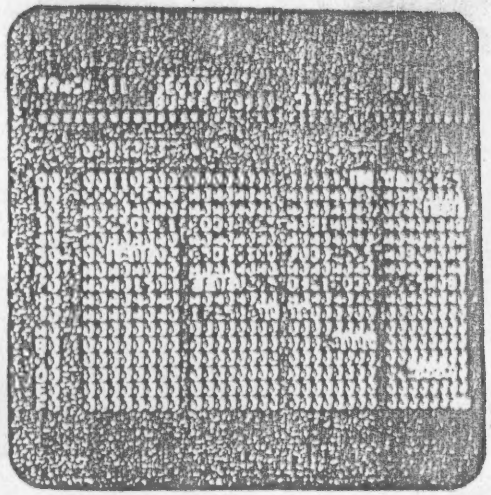

Figure 11

Figures 10 and 11 show the sectors of the diskette directory, in both HEX and ASCII mode. The file named EXAMPLE 2 has an embedded CTRL—P located between the M and P (this is why it appears to be EXAMPPLE 2 in the figure). To correct this, enter the ASCII mode and:

1. Press "E" and EDIT the sector starting at byte $54. Replace the name containing the control character (I would recommend EXAMPLE 2).
2. Press CTRL—W and write this sector back to the disk and the control character will no longer be in the filename.

UNDELETE PROGRAMS

At one time or another; almost everyone has accidentally deleted a program. Using the INSPECTOR, it is possible to recover these deleted programs. This is possible because the delete function modifies the file name but leaves the file information intact.

NOTE: THIS MAY NOT BE POSSIBLE IF ANY OTHER DATA HAS BEEN SAVED TO THE DISK SINCE THE PROGRAM WAS DELETED. THIS IS BECAUSE DOS FREES THE SECTORS OF A DELETED FILE ON THE TRACK BIT MAP WHEN IT DELETES A FILE. IF MORE INFORMATION IS SAVED TO THE DISK, IT MAY WRITE OVER THE SECTORS OF THE DELETED FILE.

In figures 10 and 11, the file named EXAMPLE 3 has been deleted. As explained on page 130 of the DOS MANUAL, when a file is deleted:

1. An $FF is placed in the location where the Track number for the Track Sector List is usually located.
2. The Track information is placed in the last location for the filename.

To undelete the file:

1. Press "E" to select the EDIT mode.
2. Put the track information, from the last location of the filename, back in its original location in place of $FF.
3. Place $A0 in the last byte of the filename (A0 is a blank space).

In our example, byte $74 is the location of the Track Sector List and byte $94 is the last byte of the filename. In the Edit mode, place a $17 (the contents of location $94) at $74. Then place an $A0 at location $94, and the file is now back in the catalog.

We must reserve the sectors of the file on the Track Bit Map again before the job is complete. Do this as follows:

1. Load the file.
2. Delete the file.
3. Resave the file to the disk.

This will reserve the sectors so that they cannot be written over.

NOTE: If the file is a text file, FID it to another disk and then FID it back again to its original disk. Do not FID it back from the original directly back to the original disk.

EDITING BASIC PROGRAMS

There are often times when it would be useful to edit BASIC programs in memory. Up until now, this has not been easy, but with the use of THE INSPECTOR, it is quite simple. It is possible to enter illegal commands, edit print statements, enter lower case characters and also enter illegal line numbers. Let's go through an example to see how this is done.

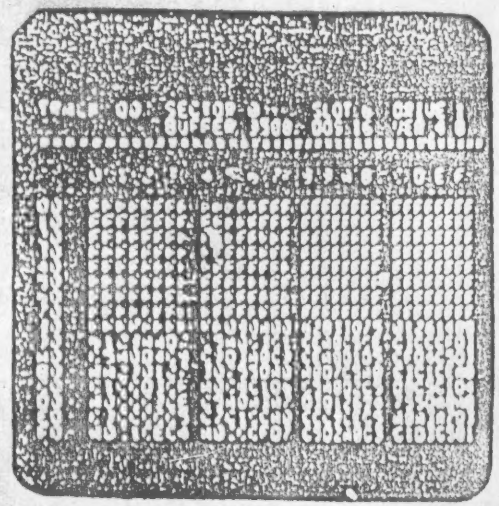

Figure 12

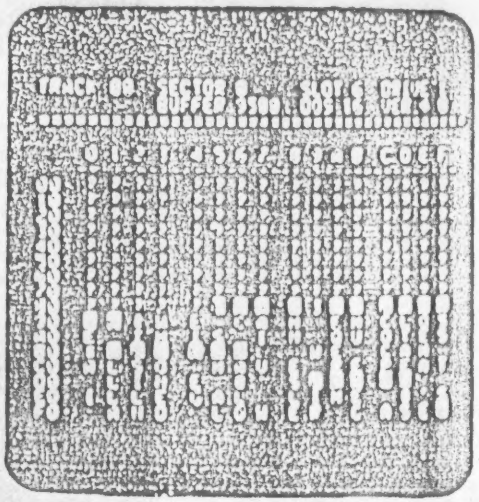

Figure 13

Figures 12 and 13 show a HEX and ASCII display for the memory location containing the following program. This location was easily found by searching for the string "WE CAN" with the F command.

```
   10 PRINT 30000
   20 PRINT "*WE CAN ENCLOSE THIS IN QUOTES*"
30000 REM WE CAN ALSO HAVE REPEAT LINE #'S
30100 REM ILLEGAL LINE #'S AND LOWER CASE
```

It is not possible to enter certain commands within a program from INTEGER BASIC. Both HIMEM: and LOMEM: are two such commands. Sometimes it is advantagous to add these commands. With THE INSPECTOR, this can be easily done.

It is a good idea to place these commands at the beginning of the program in order to easily locate the area of memory that you wish to change. In our example, we are going to change line #10 from "PRINT 30000" to "LOMEM:30000". To do this, the $62 (which is the token for PRINT) in byte $88 will be changed to $11 (the token for LOMEM:) using the Edit mode. This 1 change is all that is necessary.

Similiarly, the *'s in line #20 may be changed to "s, and the ASCII characters in line #30100 may all be changed to lower case. This may be all done with the EDIT command in the ASCII mode.

Another feature that is often seen in commercial programs is the addition of illegal or repeat line numbers at the end of a program. This is a good place to put REM statements containing a copyright notice or an authors name. In this example, the normal line numbers are located in the 2 bytes previous to the 5D's located at $B6 and $DD (the 5D is the token for REM). By placing $FF's in these locations, we can not only add line #65535, which is not normally usable, but we can also have as many as we want.

After making all of these changes, the program listing now becomes:

```
   10 LOMEM:30000
   20 PRINT ""WE CAN ENCLOSE THIS IN QUOTES""
65535 REM WE CAN ALSO HAVE REPEAT LINE #'S
65535 REM illegal line #'s and lower case.
```

AVOIDING THE RELOADING OF THE LANGUAGE CARD PROBLEM

There is an annoying problem with DOS 3.3. Whenever it is booted, either the first time or turned on by PR#6, it clobbers whatever BASIC that was loaded into the Language Card. This can be prevented by changing 1 byte in DOS. You can make this change as follows.

1. Toggle the HEX/ASCII display mode to Hexadecimal mode using the "A" key.
2. Insert your DOS 3.3 disk into the disk drive.
3. Select Track $0 using the "T" command.
4. Select Sector $9 using the "S" command.
5. Read the sector into the buffer using the "R" command.
6. Enter the Edit mode using the "E" command.
7. Select byte $CC and press (SPACE BAR).
8. Type in "10", (RETURN). This writes a $10 in byte $CC.
9. Write this sector back to the disk using the "CTRL–W" command.

You have now modified DOS on the disk so that the BASIC language will not be blown every time you boot a disk.

A BIT ABOUT NIBBLES

This section is intended to help acquaint the user with the method in which the DISK II stores and retrieves data. It is of a highly technical nature and is intended for the advanced user.

Information is stored (and retrieved) in parcels of data which we will call nibbles. Each nibble (sometimes referred to as a "disk byte") consists of 8 bits of data. These 8 bits are serially written to and read from the disk through a shift register contained in the disk controller card. The high order bit of the shift register is used to indicate to the software that an entire nibble has been read. Because of this hardware constraint, it is necessary for the high order bit of all nibbles to be set. This means that there are less than 8 significant bits in a nibble and more than 256 nibbles must be combined to represent 256 bytes of data. On a 13 sector disk, 410 nibbles are required to represent 256 bytes (one sector of data). On a 16 sector disk, 342 nibbles are required. There are routines within DOS which convert information between nibbles and data bytes.

Each sector on a track consists of two fields which are separated from each other and from adjacent sectors by gaps of special nibbles. The first field is the address field and includes the volume, track and sector information. The second field is the data field and contains the actual sector data. When a sector is read from a disk, DOS searches for a specific address field. When the desired address field is found, the data field which immediately follows is read into memory.

Each address and data field is preceeded by a unique prefix and followed by a suffix. The nibbles which make up the prefix have been chosen to insure that they do not appear elsewhere in the data. These prefixes and suffixes are commonly referred to as markers and trailers.

The address field appears as follows:

ADDRESS FIELD MARKER—D5 AA B5 (13 sector) or D5 AA 96 (16 sector)
VOLUME NUMBER (2 nibbles)
TRACK NUMBER (2 nibbles)
SECTOR NUMBER (2 nibbles)
ADDRESS FIELD CHECKSUM (2 nibbles)
ADDRESS FIELD TRAILER--DE AA

The two nibble combinations for the volume, track, sector, and checksum information are encoded using a simple algorithm. The following table is included to assist in decoding the sector number.

NIBBLES	VALUE	NIBBLES	VALUE
AA AA	00	AE AA	08
AA AB	01	AE AB	09
AA AA	02	AF AA	0A
AB AB	03	AF AB	0B
AA AE	04	AE AE	0C
AA AE	05	AE AF	0D
AB AE	06	AF AE	0E
AB AF	07	AF AF	0F

The data field appears as follows:

DATA FIELD MARKER—D5 AA AD
DATA NIBBLES (either 410 or 342)
DATA FIELD CHECKSUM (one nibble)
DATA FIELD TRAILER—DE AA

Note that the markers and trailers are different on some disks as a method of protection. If an attempt is made to read a sector on one of these disks while a normal DOS is resident in memory, an I/O ERROR will occur. If you wish to read this type of disk, please refer to the topic FOREIGN DOS in the section of this manual entitled TIPS FOR THE USER.

As mentioned earlier, the fields are separated by gaps of special nibbles which are called self—sync or auto—sync nibbles. The purpose of these nibbles is to synchronize the shift register to the start of each nibble. Without self—sync, the disk controller could not determine where each non--self—sync nibble began. Normally, self—sync nibbles appear as FF's, although some software uses self—sync other than FF.

serially written to and read from the disk through a shift register on the controller card. The high order bit of the shift register is used to indicate to the software [illegible] hardware constraint, it is necessary

SUGGESTED READING

1. Black, Preston R., Nibble, Vol 2. No. 2. 1981.
2. The DOS Manual by APPLE COMPUTER INC., 1980
3. Meador, Lee.,"Apple DOS Articles", Fort Worth Apple Users Group Newsletters, 1980.
4. Pump, Mark,"Automatic Free Space with Catalog" Call Apple, Sept. 1980, Aug. 1980.
5. Pump, Mark, "Mark Pumps DOS Patches," Harvest, NSAUG Newsletter, Oct. 1980.
6. Reynolds, William. "Disassembling DOS 3.2", Micro, Oct. 1979.
7. Reynolds, William, "Nibble", Vol. 2, no. 2, 1981
8. Worth, Don & Lechner, Pieter, "Beneath Apple DOS", Quality Software, 1981.

ADDRESSES

1. Fort Worth Apple Users Group, 1401 Hillcrest, Arlington, TX, 76010
2. "Harvest", Northwest Suburban Apple Users Group, now Northern Illinois Apple Users Group, Box 787, Palatine, IL, 60067.
3. Call Apple, Apple Pugetsound Program Library Exchange, 304 Main Ave. S. Suite 300, Benton, WA, 98055
4. Nibble, Box 325, Lincoln, MA 01773
5. Quality Software, 6660 Reseda Blvd., Suite 105, Reseda, CA 91335
6. Micro, 34 Chelmsford, Chelmsford, MA 01824

[illegible] separated by gaps of special nibbles which are called

APPENDIX
DECIMAL TO HEXADECIMAL CONVERSION TABLE

DEC.	HEX	DEC.	HEX	DEC.	HEX	DEC.	HEX	DEC.	HEX
0	00	51	33	103	67	155	9B	207	CF
1	01	52	34	104	68	156	9C	208	D0
2	02	53	35	105	69	157	9D	209	D1
3	03	54	36	106	6A	158	9E	210	D2
4	04	55	37	107	6B	159	9F	211	D3
5	05	56	38	108	6C	160	A0	212	D4
6	06	57	39	109	6D	161	A1	213	D5
7	07	58	3A	110	6E	162	A2	214	D6
8	08	59	3B	111	6F	163	A3	215	D7
9	09	60	3C	112	70	164	A4	216	D8
10	0A	61	4D	113	71	165	A5	217	D9
11	0B	62	3E	114	72	166	A6	218	DA
12	OC	63	3F	115	73	167	A7	219	DB
13	0D	64	40	116	74	168	A8	220	DC
14	0E	65	41	117	75	169	A9	221	DD
15	0F	66	42	118	76	170	AA	222	DE
16	0F	67	43	119	77	171	AB	223	DF
16	10	68	44	120	78	172	AC	224	E0
17	11	69	45	121	79	173	AD	225	E1
18	12	70	46	122	7A	174	AE	226	E2
19	13	71	47	123	7B	175	AF	227	E3
20	14	72	48	124	7C	176	B0	228	E4
21	15	73	49	125	7D	177	B1	229	E5
22	16	74	4A	126	7E	178	B2	230	E6
23	17	75	4B	127	7F	179	B3	231	E7
24	18	76	4C	128	80	180	B4	232	E8
25	19	77	4D	129	81	181	B5	233	E9
26	1A	78	4E	130	82	182	B6	234	EA
27	1B	79	4F	131	82	183	B7	235	EB
28	1C	80	50	132	84	184	B8	236	EC
29	1D	81	51	133	85	185	B9	237	ED
30	1E	82	52	134	86	186	BA	238	EE
31	1F	83	53	135	87	187	BB	239	EF
32	20	84	54	136	88	188	BC	240	F0
33	21	85	55	137	89	189	BD	241	F1
34	22	86	56	138	8A	190	BE	242	F2
35	23	87	57	139	8B	191	BF	243	F3
36	24	88	58	140	8C	192	C0	244	F4
37	25	89	59	141	8D	193	C1	245	F5
38	26	90	5A	142	8E	194	C2	246	F6
39	27	91	5B	143	8F	195	C3	247	F7
40	28	92	5C	144	90	196	C4	248	F8
41	29	93	5D	145	91	197	C5	249	F9
42	2A	94	5E	146	92	198	C6	250	FA
43	2B	95	5F	147	93	199	C7	251	FB
44	2C	96	60	148	94	200	C8	252	FC
45	2D	97	61	149	95	201	C9	253	FD
46	2D	98	62	150	96	202	CA	254	FE
47	2F	99	63	151	97	203	CB	255	FF
48	30	100	64	152	98	204	CC		